REMARKS

Claims 1-26 are pending and under consideration. Claims 1-26 were rejected in the Final Office Action mailed October 12, 2005. In this Amendment, no claims are added or canceled. Claims 1 and 13 are amended. These amendments are supported by the specification, for example, at paragraph 21 and paragraphs 123-139.

I. Rejection of claims 1-4, 6-15, and 17-25 under 35 U.S.C. §103(a) as being allegedly unpatentable over *Calder (U.S. Patent No. 5,963,972) in view of Lomet (U.S. Patent No. 5,963,972)*:

Applicants respectfully traverse the rejection, and respectfully submit that he combination of *Calder* and *Lomet* fails to teach or suggest all of the limitations of independent claims 1, 8, 12, 19, 20, 23, and 24.

Regarding claim 1, the combination of *Calder* and *Lomet* fails to teach or suggest, for example, facilitating development of the data flow program by generating a graph representing the blocks and the determined dependencies and displaying the graph to a user. The Examiner asserts that *Calder* teaches displaying such a graph, and refers to column 3, lines 56-66, column 7, lines 6-56, column 8, lines 10-54, and column 12, lines 25-37 of *Calder*. Contrary to the Examiner's assertion none of those sections of *Calder*, nor any other section of *Calder*, actually teaches or suggests displaying a graph to a user. Because *Calder* is directed to a code reordering algorithm for optimizing a cache and does not require human input or intervention, there is no reason for displaying *Calder*'s graph to a user. In fact *Calder* does not even mention the words "display" or "visualize" in entire description.

Moreover, *Calder* fails to teach or suggest facilitating development of a data flow program by any means. *Calder* is directed to a code reordering method for minimizing cache conflicts. (See column 9, lines 21-26 of *Calder*). Thus, *Calder* does not facilitate program development, and is instead directed to cache optimization. Furthermore, the programs in *Calder* are control flow programs, not data flow programs. (See column 3, lines 56-62 of *Calder*). The graph of a control flow program graph is different in representation from a data flow program graph, and presents a problem of an entirely different nature. (See paragraphs 9-18 of the present application).

For at least these reasons, *prima facie* obviousness has not been established, and claim 1 is patentable over the combination of *Calder* and *Lomet*. Claims 8, 12, 20, 23, and 24 also recite displaying or visualizing a graph for a data flow program, and are therefore patentable for at least the same reasons.

Regarding claim 2, the combination of *Calder* and *Lomet* fails to teach or suggest a graph comprising nodes assigned to the blocks and dependency arcs representing the determined dependencies. *Calder* creates a graph that has nodes and edges. The nodes represent units of instructions of a program and the edges represent execution relationships between the nodes. (See column 3, lines 56-62 of *Calder*). Thus, unlike Applicants' claimed invention in which graph nodes are assigned to blocks of memory, *Calder's* graph nodes merely correspond to units of instructions -- the units of instructions do not correspond to blocks of memory.

After *Calder* creates a graph, *Calder* assigns weights (based on popularity) to the graph's edges. Then, graph nodes that are connected by higher-weight edges are assigned to cache memory before graph nodes that are connected by lower-weight edges. (See column 4, lines 41-58 of *Calder*). Therefore, the instructions that are in graph nodes that are connected by higher-

weight edges can be optimally placed in the cache. Accordingly, *Calder's* graph nodes do not correspond to blocks of memory. Instead, the blocks merely contain instructions, which are assigned to cache memory after the graph has already been created.

For at least these reasons, *prima facie* obviousness has not been established, and claim 2 is patentable over the combination of *Calder* and *Lomet*. Claim 9 recites similar limitations and is therefore patentable for at least the same reasons.

Regarding claims 3, 4, 6, and 7, the Examiner contends that these limitations are taught by *Calder* at column 5, lines 36-49, which states:

> The weights 103 of the edges 102 represent the number of times each edge is traversed. The sum of the weights of the edges entering and exiting a node indicate the number of incoming and outgoing calls, and therefore the relative "popularity" of an instruction unit. Other weighing heuristics can also be used.
>
> After the flow graph 100 has been built, the popularity of each unit is considered. Based on the popularity, the flow graph 100 is partitioned into popular and unpopular sets of units and edges. The popular instruction unit set includes as members all units which are frequently a caller or callee, and the popular edge set includes frequently traversed edges. All other instruction units and edges are assigned to unpopular sets.

Applicants respectfully submit that this passage clearly fails to teach or suggest any of presenting a dependency arc using an unsatisfied dependency visualization (claim 3), displaying in a visually distinctive manner unmet dependencies in a graph (claim 4), displaying unexecuted nodes using an unexecuted visualization (claim 6), and accentuating a portion of a data structure accessed by a code segment (claim 7). *Calder* fails to teach displaying a graph at all, let alone displaying unsatisfied dependencies differently from satisfied dependencies.

Accordingly, *prima facie* obviousness has not been established, and claims 3, 4, 6, and 7 are patentable over *Calder* and *Lomet*. Claims 9-11, 14, 15, 17, 18, 21, and 22 recite similar limitations, and are therefore patentable for at least the same reasons.

Regarding claim 19, *Calder* and *Lomet* fail to teach or suggest displaying a graph, as previously discussed. Furthermore, the combination fails to teach or suggest "while the code segments are executing, determining which nodes in the graph are unexecuted nodes and which nodes in the graph are executed nodes; and displaying the unexecuted nodes in a manner visually distinctive from the executed nodes." Applicants respectfully submit that neither *Calder* nor *Lomet* teach or suggest displaying unexecuted nodes while the code segments are executing. The Examiner again points to column 5, lines 36-49 of *Calder*; however, the portion of *Calder*, as evidenced above, provides no suggestion for displaying the unexecuted nodes in a manner visually distinctive from the executed nodes while the code segments represented in the graph are executing. Accordingly, *prima facie* obviousness has not been established, and claims 19 is patentable over *Calder* and *Lomet*.

II. Rejection of claims 5, 16, and 26 under 35 U.S.C. §103(a) as being allegedly unpatentable over *Calder (U.S. Patent No. 5,963,972)* in view of *Lomet (U.S. Patent No. 5,963,972)*, and further in view of *Cai (U.S. Patent No. 6,349,363)*:

Applicants respectfully traverse the rejection.

Claims 1, 12, and 24 are allowable as discussed above. *Cai* still fails to disclose or suggest Applicants' claimed data read and data write identifiers and fails to disclose or suggest determining dependencies based on the data read and data write identifiers. Therefore, *Calder* in view of *Lomet* and *Cai* still fails to disclose or suggest claims 1, 12, and 24.

Claims 5, 16, and 26 depend directly or indirectly from claims 1, 12, or 24 and are therefore allowable for at least the same reasons that claims 1, 12, and 24 are allowable.

CONCLUSION

In view of the foregoing, it is submitted that claims 1-26 are patentable. It is therefore submitted that the application is in condition for allowance. Notice to that effect is respectfully requested.

Respectfully submitted,

Date: February 13, 2006

A. Wesley Ferrebee
Reg. No. 51,312

Customer No. 58328
SONNENSCHEIN NATH & ROSENTHAL LLP
P. O. Box 061080
Wacker Drive Station - Sears Tower
Chicago, Illinois 60606-1080
Telephone (202) 408-6832
Facsimile: (312) 876-7934